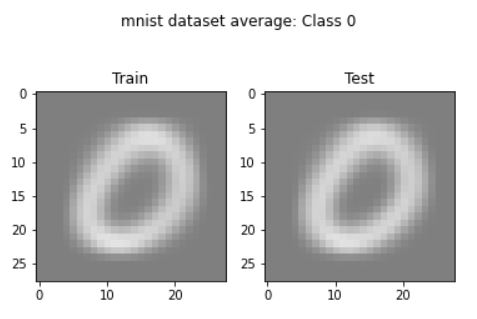
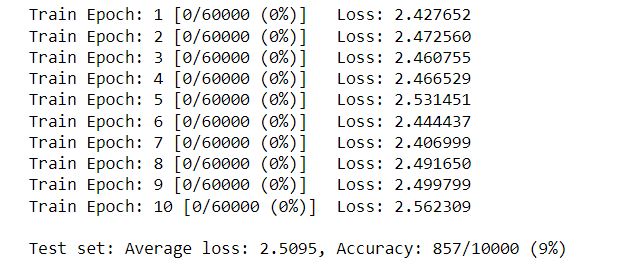
Home Work 3

Ex 1:

* Q-1: for this question we calculated the average image for each class in MNIST both for the training data set and the test one. After that we normalized the values in the averaged images By finding the MIN and MAX values in each image.  
  Then, Using MATPLOT we ploted the results as follows:

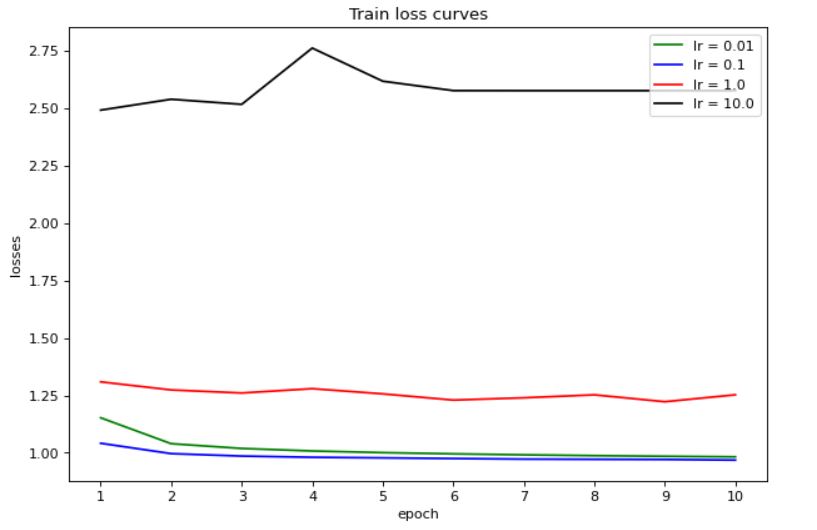


* Q-2:   
  A) By using Pytorch utilty we created the dataloader objects to load the data and process it, then we trained the CNN and plotted the resulted included the loss and eventually we tested it, and plotted the accuracy.   
  After that we retrieved the weights from the network parameters, and plotted them as normalized images.  
    
  B) We created an new method and we call it limited\_train() wich reduced the number of traning examples to just 50, then we used it for training our new model and got the following results:



Explanation: As we can see “To be continue” -under fitting

* Q3:  
  A) We defined a New Class called MultiLayerNet() in which we added a new hidden with 1000 Units. In the Forward Function, we called the mentioned layers and applied the tanh nonlinearity.  
    
  B) We created a new Model unsing the mentions Class and trained it on MNIST 4 times with different learning rate each time [ 0.01 , 0.1 , 1.0 , 10.0 ] followed with testing it.  
  Eventually we Plotted the training loss curve for each learning rate and got the following results:



For the test accuracy, the learning rate 0.1 and the 0.01 got the best results and they were so close, that’s because **outliers** less impact on the network than in the other layers.   
The same reason explains why we got better results in learning rate 1 than in learning rate 10.

- Q-4:

when we used the 3x3 convolution Layers, We got the following output:

